**EXPERIMENT 3**

**IMPLEMENTATION OF MULTI-ARMED BANDITS USING OPEN GYM AI API**

Code

import gym

import gym\_bandits

import numpy as np

import warnings

import matplotlib.pyplot as plt

warnings.filterwarnings("ignore")

env = gym.make('MultiarmedBandits-v0', disable\_env\_checker=True)

n\_arms = env.action\_space.n

num\_episodes = 500

epsilon = 0.1

q\_values = np.zeros(n\_arms)

counts = np.zeros(n\_arms)

reward\_sums = np.zeros(n\_arms)

all\_rewards = []

action\_history = []

for episode in range(num\_episodes):

env.reset()

if np.random.rand() < epsilon:

action = env.action\_space.sample() # Explore

else:

action = np.argmax(q\_values) # Exploit

observation, reward, done, info = env.step(action)

counts[action] += 1

reward\_sums[action] += reward

q\_values[action] = reward\_sums[action] / counts[action]

all\_rewards.append(reward)

action\_history.append(action)

env.close()

print("Estimated Q-values for each arm:", np.round(q\_values, 2))

print("Number of times each arm was selected:", counts.astype(int))

print("Average reward over all episodes:", round(np.mean(all\_rewards), 3))

plt.figure(figsize=(14, 5))

plt.subplot(1, 2, 1)

plt.plot(np.cumsum(all\_rewards), color='blue')

plt.title("Cumulative Reward Over Episodes")

plt.xlabel("Episode")

plt.ylabel("Cumulative Reward")

plt.subplot(1, 2, 2)

plt.bar(np.arange(n\_arms), counts, color='green')

plt.title("Action Selection Frequency")

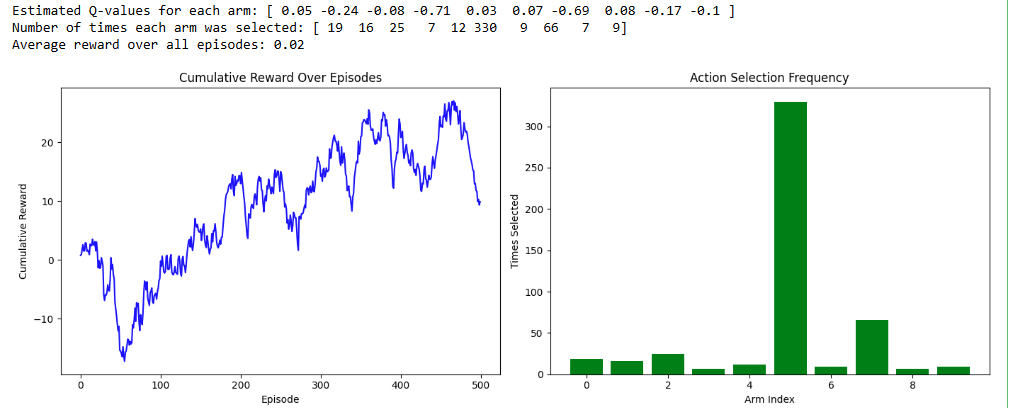
plt.xlabel("Arm Index")

plt.ylabel("Times Selected")

plt.tight\_layout()

plt.show()

Output



E0123038

MUGESH K